AEROFAST: Functional and Real-Time Simulation for Aerocapture GNC Assessment

Miguel Hagenfeldt[(1)], *Andrea Mafficini*[(1)], *Vicente Fernandez*[(1)] , *Sergio Gómez*[(1)] , *Carlos Valle*[(1)] , *Pedro Palomo*[(1)] , *Antonio Latorre*[(1)]

DEIMOS Space, S.L.U.[(1)]
Ronda de Poniente, 19. Edificio Fiteni VI, 2-2
28760 Tres Cantos (Madrid) – Spain
E-mail: vicente.fernandez@deimos-space.com

ABSTRACT

The paper describes the AEROFAST simulation facilities integrated into an unified simulator architecture to support functional (FES), processor-in-the-loop (PIL) and hardware-in-the-loop (HIL) simulation needs. The unified simulator architecture features a layered approach to the simulators architecture and life-cycle, maximizing the reuse of models/functions and bridging the gaps between the non-real-time and real-time engineering domains. Consequently it offers consistency in the employed models in FES/PIL/HIL and provides means for rigorous sequential and regression testing of GNC algorithms. The paper also addresses the lessons learned from the development and usage of the AEROFAST simulation tools during the execution of the simulation campaigns. The campaigns, designed to validate the performance of the GNC algorithms, comprised nominal, Monte Carlo and real-time nominal and (execution time) worst cases. In particular, the unified approach was found to be very useful for the encountered design loops during non-real-time and real-time assessments.

INTRODUCTION

AEROFAST, which stands for “Aerocapture for Future Space Transportation” [1], is a collaborative project funded by the European Commission through the Seventh Framework Programme (FP7). It aims at raising the maturity of the aerocapture technology in Europe in preparation for the future exploration of Mars. For this purpose, a demonstration mission was devised, consisting of an interplanetary transfer with radiometric/optical navigation followed by an aerocapture manoeuvre on Mars. Within this context, DEIMOS Space developed the simulation facilities for the assessment of the GNC algorithms designed for the pre-aerocapture and aerocapture phases of the mission.

The non-real-time AEROFAST simulators are Functional Engineering Simulator (FES) facilities based on the DEIMOS’ SIMPLAT reusable simulation infrastructure for MATLAB/Simulink. Actually, different FES facilities were developed for the two mission phases, namely the pre-aerocapture FES (to validate the DEIMOS GNC algorithms) and the aerocapture FES (to validate Astrium Space Transportation GNC algorithms). The real-time AEROFAST simulators are Functional Validation Testbench (FVT) facilities with PIL and HIL configurations.

The verification and validation (V&V) approach respects the standards specified in ECSS-E-TM-10-21A, ECSS-E/Q-ST-40C/80C and ECSS-E-60A. Within the complete simulation environment the V&V is divided in a two-fold iterative process, as shown in Fig. 1. The FES simulation process only includes the top and bottom sub-processes.

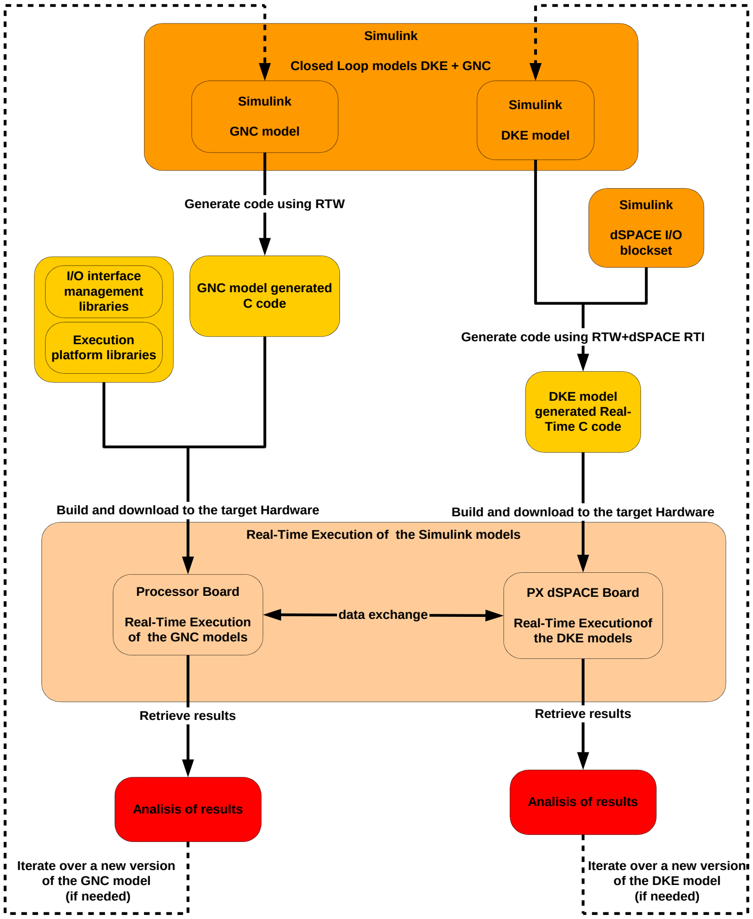

Simulink
Closed Loop models DKE + GNC
Simulink
GNC model
Simulink
DKE model
Generate code using RTW
Simulink
dSPACE I/O blockset
I/O interface management libraries
Execution platform libraries
GNC model generated C code
Generate code using RTW+dSPACE RTI
DKE model generated Real-Time C code
Build and download to the target Hardware
Build and download to the target Hardware
Real-Time Execution of the Simulink models
Processor Board
Real-Time Execution of the GNC models
data exchange
PX dSPACE Board
Real-Time Executionof the DKE models
Retrieve results
Retrieve results
Analisis of results
Analisis of results
Iterate over a new version of the GNC model (if needed)
Iterate over a new version of the DKE model (if needed)

Fig. 1. Complete development and simulation process

To achieve the devised process(es) an unified architecture of the FES/PIL/HIL infrastructure is deemed essential. It allows both the simulation and GNC engineers to perform sequential (nominal flow) and regression testing (in case of SW changes). The following paragraphs detail the architectural and practical aspects of this unified approach.

UNIFIED ARCHITECTURE ELEMENTS

There are two main implementations for the simulation infrastructure: non-realtime (NRT) and realtime (RT). The NRT implementation consists on the Functional Engineering Simulators (FES), for both Pre-aerocapture and Aerocapture mission phases, consisting each of a “Real World” simulator and GNC algorithms running in a Simulink/MATLAB runtime environment. The RT implementation is further sub-divided into two configurations: processor in-the-loop (PIL) and hardware in-the-loop (HIL) configurations. The relationship between these implementations/configurations relies on a sequential development/validation process having 1) the PIL based as much as possible on the FES architecture, for both Pre-aerocapture and Aerocapture mission phases, and 2) the PIL architecture serving, for the Pre-aerocapture mission phase, as a basis for the HIL configuration. This relationship is depicted in Fig. 2. The following sections provide brief details on these implementations with the intention of providing better understanding of the unified architecture.

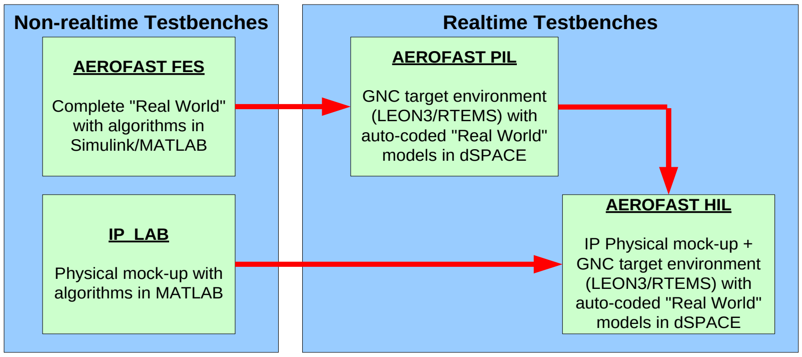

Non-realtime Testbenches
Realtime Testbenches
AEROFAST FES
Complete "Real World" with algorithms in Simulink/MATLAB
AEROFAST PIL
GNC target environment (LEON3/RTEMS) with auto-coded "Real World" models in dSPACE
IP_LAB
Physical mock-up with algorithms in MATLAB
AEROFAST HIL
IP Physical mock-up + GNC target environment (LEON3/RTEMS) with auto-coded "Real World" models in dSPACE

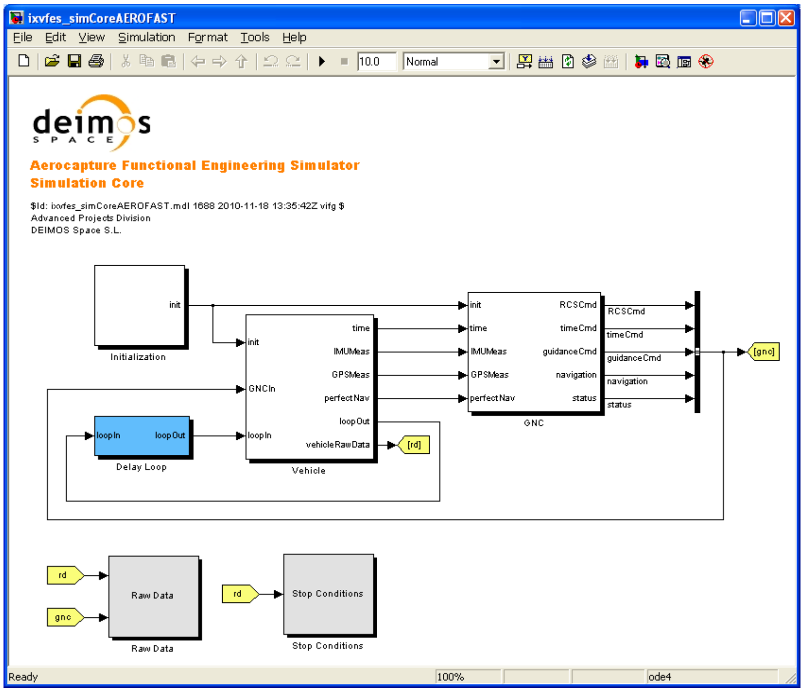
Fig. 2. Development stages for simulation facilities

Non-realtime functional engineering simulators

The Pre-aerocapture FES is a 4-DOF simulator that provides modelling of translational (3 DOF) dynamics/kinematics and attitude kinematics via a performance model (stated here as 1DOF). It supports the AEROFAST pre-aerocapture phase GNC development/verification, which in turn provides means for trajectory control from exit of Earth orbit until arrival to Entry Interface Point (EIP) near Mars. The simulator includes the necessary features to feed the GNC with navigation data, i.e. radiometric measurements (Ground Link), navigation cameras observables and ADCS estimation data. It implements celestial ephemerides information and S/C translational dynamics/kinematics environment that is subject to external perturbations and “Main propulsion”.

The Aerocapture FES is a true 6-DOF re-entry simulator and is specifically designed to support the performance assessment and validation of the Aerocapture GNC algorithm, which must control the trajectory of a biconic shape capsule during the aerocapture phase around Mars, from the Entry Interface Point (EIP) to the atmospheric exit. The simulator is derived from the IXV FES [2] and supports two types of simulations: 1) 3-DOF simulation, mainly intended for guidance testing, in which only the translational dynamics is simulated, considering the external forces acting on the vehicle, but not the external moments; 2) 6-DOF simulation, intended for evaluation of the complete GNC, in which both translational and rotational dynamics are propagated taking into account external forces and moments. The Aerocapture FES includes models for 3- and 6-DOF dynamics/kinematics, atmosphere and respective dispersions, gravity field, biconic sled shape aerodynamics, mass centring and inertia, reaction control system and inertial measurement unit. The Simulink model is shown in Fig. 3.

Both simulators are based on DEIMOS SIMPLAT reusable simulation infrastructure for MATLAB/Simulink, which offers a man-machine interface, simulation engine, raw data management, post-processing and plotting functionalities. Moreover, it provides the necessary resources to run single-shot and Monte Carlo simulations and to compute GNC performance index statistics so that the analyst can verify the compliance to GNC requirements. The customisation for each project is based on Configuration Data Files (XML files) and specific post-processing and plotting functions.



ixvfes_simCoreAEROFAST

Aerocapture Functional Engineering Simulator
Simulation Core

$Id: ixvfes_simCoreAEROFAST.mdl 1688 2010-11-18 13:35:42Z vifg $
Advanced Projects Division
DEIMOS Space S.L.

Fig. 3. Aerocapture FES Core

Realtime testbenches

As referred earlier, the RT implementation consists on two configurations: PIL and HIL.

Fig. 4 presents an overview of the PIL configuration, which includes the following features:

- PX dSPACE Box runs in Real-Time the DKE, Sensors and actuators modelled in Simulink. The PIL configuration does not include any explicit image generation and processing (done in HIL configuration) but a performance model of the Image Generation (IG) – Camera – Image Processing (IP) chain.
- Control Desk computer monitors and controls the execution of the whole experiment setup.
- LEON 3 processor runs the GNC model, connected in closed loop to the PX dSPACE Box.

The HIL configuration includes additional modules (wrt PIL configuration), with the following features:

- The HIL configuration involves a mock-up with spherical targets, lights and a camera with zoom placed on top of a pan and tilt unit.
- The HIL configuration executes IP routines in real time on the images grabbed from the camera.
- Again PX dSPACE Box runs in Real-Time the DKE, Sensors and actuators modelled in Simulink but now using image generation chain hardware in the loop. It also manages the Pan & Tilt unit and the Camera zoom.
- LEON 3 processor runs the GNC model and the Image Processing SW in real-time using as an input the images picked up by the real camera.

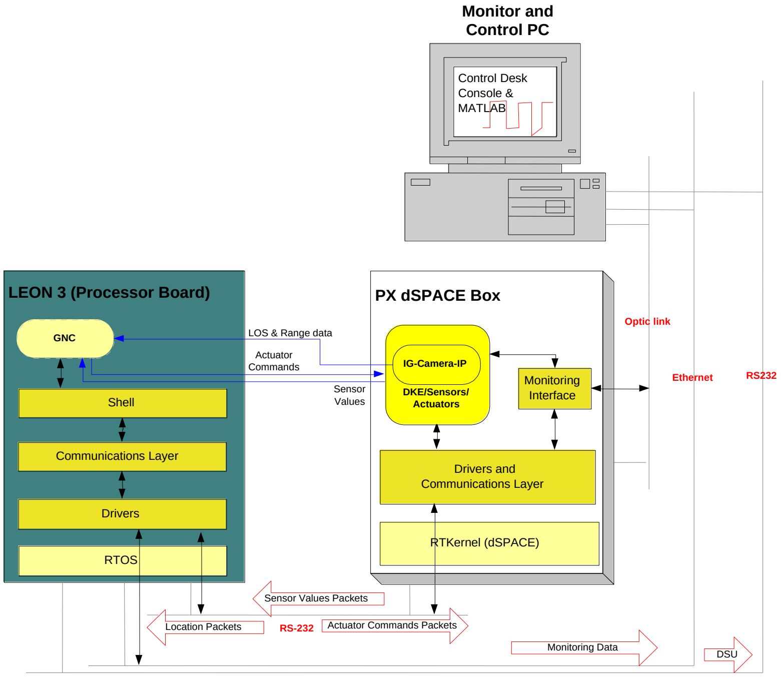


Fig. 4. PIL configuration architecture

The RT simulators DKE models are intrinsically related to the NRT simulators models, although, not all models are exactly the same. Some RT models are indeed direct auto-coded versions of the NRT models but there are exceptions. Examples are the delays, image processing and IP mock-up control blocks. The delays are only applicable in NRT. The image processing in FES/PIL is a performance model while the complete SW is entirely integrated in HIL, together with a non-empty IP mock-up control block. The method to manage these similarities and differences is explained in the following section.

Unified architecture

The employed method to achieve a transparent, compatible and user-friendly implementation is the use of models variants with well defined interfaces (of which some models and/or interfaces may be dummies). The method is compatible with MATLAB/Simulink capabilities to setup and redirect library links, so that each configuration/scenario has the correct models. The interfaces are the same, the interconnections (and models variants) are compatible but the contents of the models vary and adapt to the needs. From a user point-of-view this process is transparent and in turn very user-friendly. The diagrams shown in Fig. 5 outline the level-2 static interface diagrams for the PIL and HIL configurations, in which the referred models variants method is visible.

Another requirement is to split each simulator's architecture in two, the SW to be run in dSPACE and LEON3 machines/processors. This is achieved by having two (higher-level) libraries that are merged in a top-level Simulink model. Combining the various needs and choices the simulators need to be implemented according to a layered and modular organization of models, subsystems and libraries. It is shown in Fig. 6.

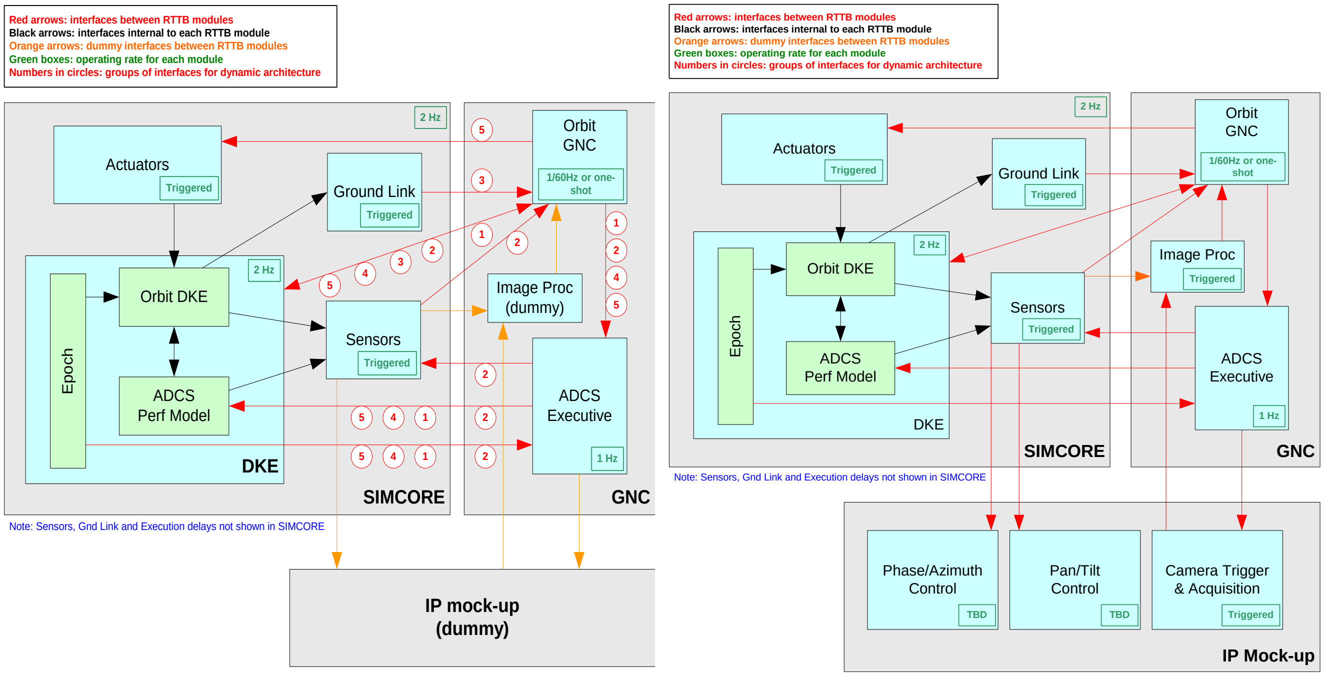

Red arrows: interfaces between RTTB modules
Black arrows: interfaces internal to each RTTB module
Orange arrows: dummy interfaces between RTTB modules
Green boxes: operating rate for each module
Numbers in circles: groups of interfaces for dynamic architecture
Actuators
Triggered
Ground Link
Triggered
2 Hz
Orbit DKE
Epoch
ADCS Perf Model
Sensors
Triggered
DKE
SIMCORE
Orbit GNC
1/60Hz or one-shot
Image Proc (dummy)
ADCS Executive
1 Hz
GNC
1
2
3
4
5
Note: Sensors, Gnd Link and Execution delays not shown in SIMCORE
IP mock-up (dummy)
Red arrows: interfaces between RTTB modules
Black arrows: interfaces internal to each RTTB module
Orange arrows: dummy interfaces between RTTB modules
Green boxes: operating rate for each module
Numbers in circles: groups of interfaces for dynamic architecture
Actuators
Triggered
Ground Link
Triggered
2 Hz
Orbit DKE
Epoch
ADCS Perf Model
Sensors
Triggered
DKE
SIMCORE
Orbit GNC
1/60Hz or one-shot
Image Proc
Triggered
ADCS Executive
1 Hz
GNC
Note: Sensors, Gnd Link and Execution delays not shown in SIMCORE
Phase/Azimuth Control
TBD
Pan/Tilt Control
TBD
Camera Trigger & Acquisition
Triggered
IP Mock-up

Fig. 5. Pre-aerocapture PIL (left) and HIL (right) level-2 static interfaces diagrams

Another relevant aspect of the unified architecture is the proper modularization of the FES and PIL/HIL setup scripts. The RT simulators re-use as much as possible the FES customised functionalities provided by the SIMPLAT infrastructure and the models variants setup scripts are shared by both implementations. In the specific case of the RT implementations, additional scripts are used to setup the RTW/dSPACE auto-coding/build and LEON3 build for each simulation. The last employs both the build of DEIMOS GNC algorithms (as C code) [3] and Astrium Space Transportation GNC algorithms (as RTEMS object files) [1]. The simulation models, GNC C code and pre-compiled binary files are stored in a shared repository location and used for both NRT and RT simulations. Thus, the unified architecture not only offers an user-friendly infrastructure but also offers strict and organised configuration control of the simulation and GNC software.

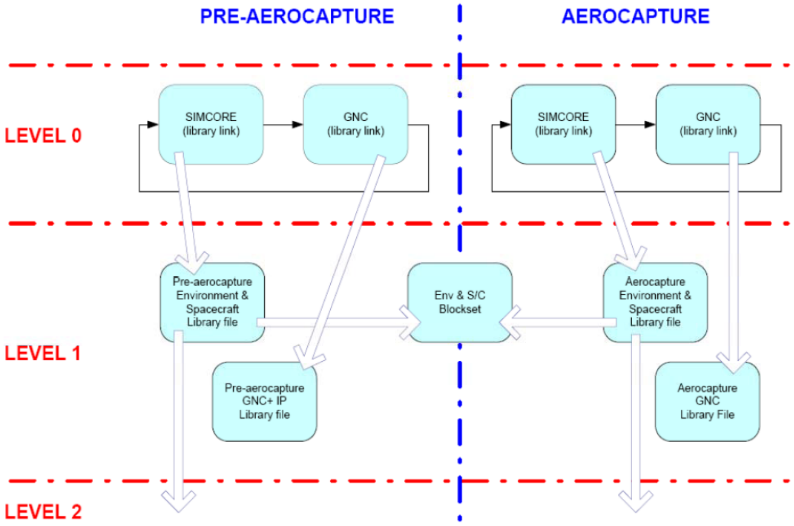

PRE-AEROCAPTURE
AEROCAPTURE
LEVEL 0
LEVEL 1
LEVEL 2
SIMCORE (library link)
GNC (library link)
SIMCORE (library link)
GNC (library link)
Pre-aerocapture Environment & Spacecraft Library file
Env & S/C Blockset
Aerocapture Environment & Spacecraft Library file
Pre-aerocapture GNC+ IP Library file
Aerocapture GNC Library File

Fig. 6. Simulators layered architecture

NON- AND REALTIME SIMULATIONS

Realtime simulations approach

There are important operational and planning constraints related to Monte Carlo (MC) simulations in a RT environment that need to be accounted for when defining the V&V approach. These simulations would be difficult to implement in dSPACE without need of operator (manual) intervention and would take unnecessary long time for execution/debugging. One must consider the fundamental difference between functional and real-time validation. The first intends to seek, as far as possible, conditions that may be adverse (apart from nominal) for the GNC algorithms under test. This means that Monte Carlo tests are appropriate. On the other hand, the RT V&V intends to verify the correct operation of the system in realistic computational and latency conditions, i.e. verify if the execution time deadlines are met in the longest code execution path.

What is the chosen approach? Run Monte Carlo simulations in FES, then identify the longest execution case, extract the initial conditions from the data logs, and use these conditions for RT V&V. How does one fetch the worst case delays for GNC code? The solution relies on profiling the GNC SW during NRT Monte Carlo runs. In summary, the approach relies on two RT simulations: one nominal (no dispersions) and one (execution time) worst-case (fetched from dispersed FES MC simulations).

Comparison of NRT and RT results

Upon start of aerocapture RT V&V some important differences between NRT and RT simulations results were identified. The plots on the left of Fig. 7 show that in the non-real-time simulation the guidance commands two bank reversal manoeuvres, whereas in the real-time simulation it commands only one bank reversal. This different behaviour of the GNC algorithm can be explained by the guidance computational delay. The (first version of) NRT simulation did not include a model of the GNC computational delays. To prove this justification a model of computational delay was included in the non-real-time FES simulation and the outputs were compared with those of the previous simulation. The results, depicted on the plots on the right of Fig. 7, show that the impact of the computational delay model on the overall performance is similar to that observed in the real-time simulation. These differences lead to an update of the NRT simulators and the repetition of simulation campaigns, which in turn were easily achieved by taking advantage of the unified architecture. This is explained further in the following section.

Sequential and regression testing

The unified architecture allowed performing sequential (normal flow) and regression testing in a simple and transparent manner. As result of the sequential testing (FES to PIL) the differences between NRT and RT stated above were identified. This fact required the aerocapture NRT simulator to include a GNC delay block so that the full Monte Carlo campaign could be repeated in order to 1) GNC requirements to be verified (in the new circumstances) and 2) new worst-case timing conditions to be extracted and RT simulations to be repeated. The final NRT and RT test results were coherent. Similarly, in the pre-aerocapture RT test campaign it was identified that some GNC SW look-up tables were occupying too much memory for the available LEON3 memory space. Regression testing was required upon reducing the size (and resolution) of the look-up tables. Again, final NRT and RT test results were coherent.

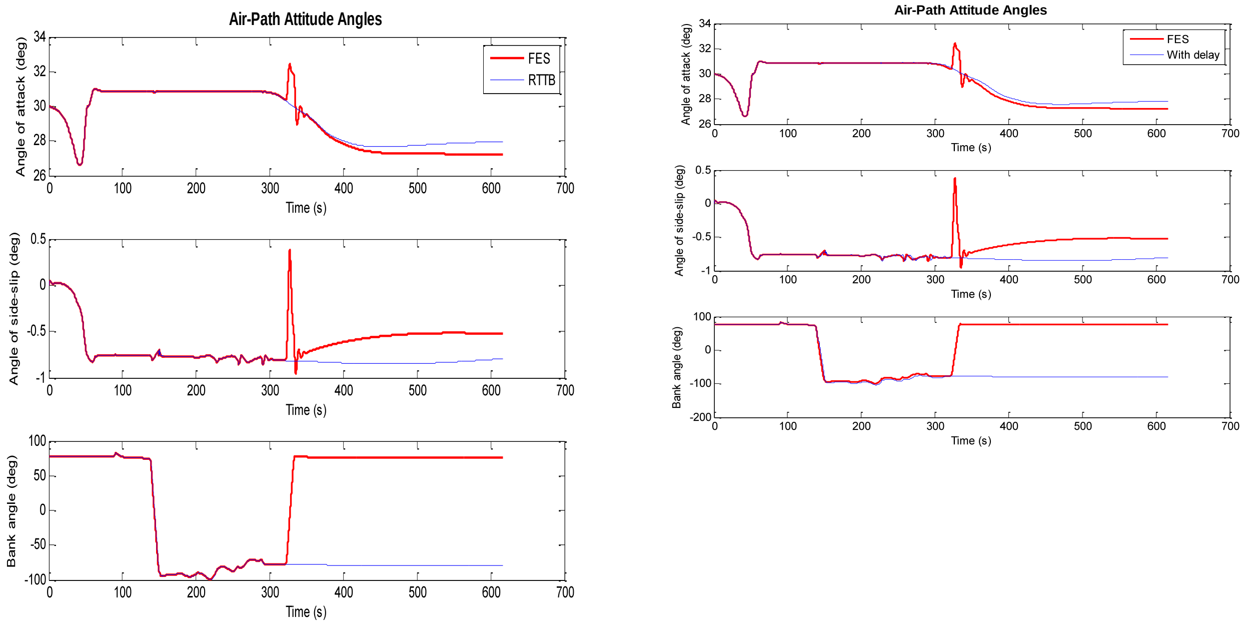


Fig. 7. Left: comparison of NRT and RT simulations results, Right: comparison of NRT simulations results with and without computational delays

CONCLUSIONS AND LESSONS LEARNED

The following conclusions and lessons learnt were derived upon development and use of the unified architecture:

- It is feasible and valuable to implement an unified FES/PIL/HIL architecture;
- It is essential that the architecture and folder structure are set-up at development kick-off, to allow proper model library and database files sharing;
- It allowed an efficient reuse of simulation models and features;
- The RT S/W implementation imposed restrictions on functional design, identified early as the unified architecture allowed quick start of RT simulation campaigns;
- It allowed easy execution of regression tests, to achieve functional revalidation when changes were required to simulator and/or GNC models.

REFERENCES

[1] AEROFAST project website: http://www.aerofast.eu

[2] V. Fernández, G. Zaiacomo, A. Mafficini, D. Mostaza, M. Kerr, L.F. Peñín, “The IXV GNC Functional Engineering Simulator,” *Proceedings of 11th International Workshop on Simulation and EGSE facilities for Space Programmes (SESP2010)*, ESA/ESTEC, Noordwijk, The Netherlands, September 2010.

[3] A. Mafficini, M. Hagenfeldt, J.L. Cano, L.F. Peñin, A. Caramagno, J.M. Rebordão, J. Dinis, “Mars Pre-aerocapture GNC design and performance,” *Proceedings of 8th International ESA Conference on Guidance, Navigation and Control Systems (GNC2011)*, Karlovy Vary, Czech Republic, June 2011.